Centrifuge Rotor Models

A Comparison of the Euler-Lagrange and the Bond Graph Modeling Approach

Jose J. Granda, Ph.D., Professor
Department of Mechanical Engineering
California State University - Sacramento

Jayant Ramakrishnan, Ph.D.
ARES CORPORATION

Louis H. Nguyen
Aeroscience & Flight Mechanics Division
NASA Johnson Space Center

AIAA-Houston Annual Technical Symposium 2006
NASA Johnson Space Center - Gilruth Center
May 19, 2006

Topics

- Objectives
- Modeling Approach Comparisons
  - Euler-Lagrange Approach
  - Bond Graph Methods
- Model Structures
  - Simple Oscillator
  - Two Mass System
  - Discussion
- Application
- Summary

Objectives

- Demonstrate the similarities and differences in traditional models and bond graph methods
- Utilize the bond graph methods for independent verification of models
- Develop and promote the bond graph modeling capability here at JSC

Acknowledgements

- We wish to acknowledge the help of Murugan Subramaniam, AkimaTechlink Systems for many helpful discussions on these topics.

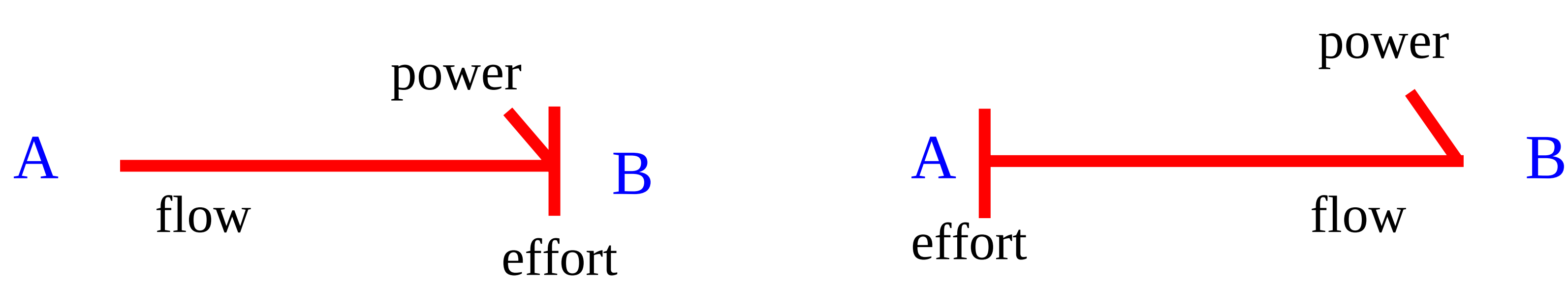
Modeling Approach Comparisons

- Dynamical Systems can be modeled in many ways. Some of the methods are:
  - Euler Lagrange Methods where the second order differential equations are derived using free body diagrams and then transformed using appropriate state variables to a system of first order equations. These systems generally have a predefined structure and their solution methods are robust and well documented.
  - Bond Graph method which is based on power and causality flow between inertial, resistive and capacitive elements and equations are directly generated in the first order state space. The choice of variables and modeling methodology results in a different structure of matrices. Programs such as CAMP-G can directly generate the equations of motion and the MATLAB M-files from the graphical input of the system structure.

Bond Graphs and Physical Variables

Power Flow Concept

Causality Concept



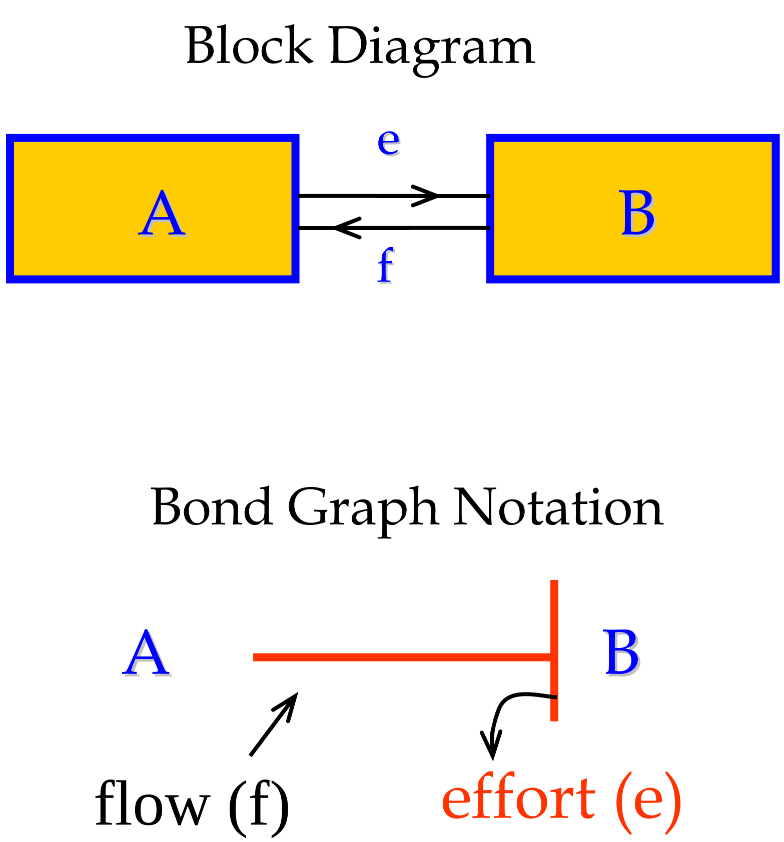
A imposes effort on B,
B responds with a flow

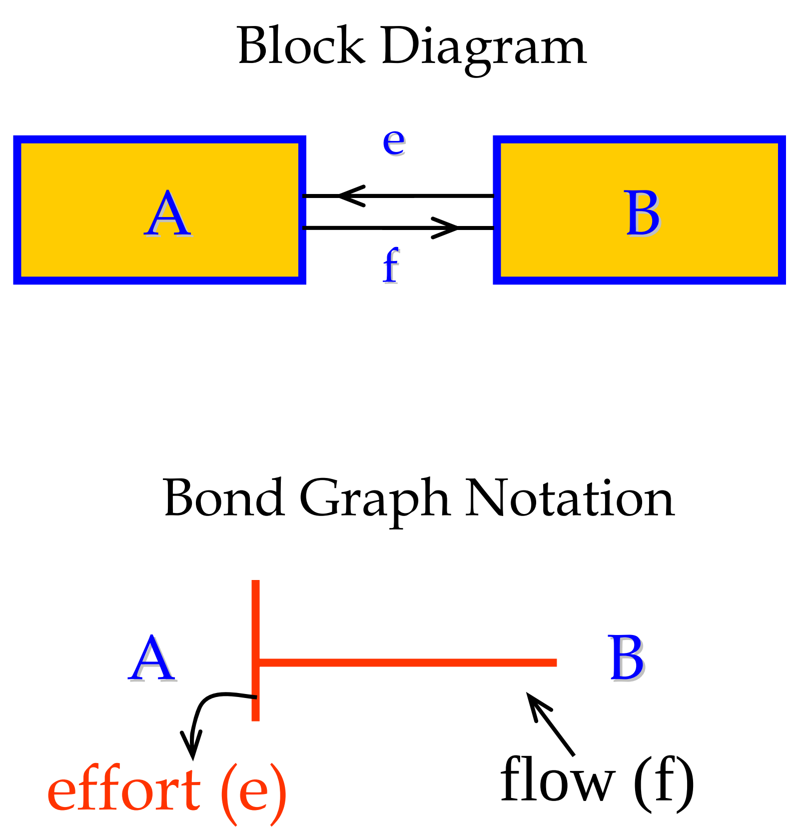
B imposes effort on A,
A responds with a flow

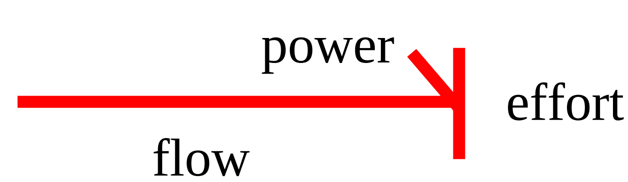
Physical Systems Variable Types

Variables	Mechanical Translation	Mechanical Rotation	Electrical	Hydraulic
Effort	Force (F) [Newtons N]	Torque (T) [N-m]	Voltage [Volts V]	Pressure (P) [N/m^2]
Flow	Velocity (v) [m/s]	Angular velocity (ω) [rad/s]	Current (i) [Amperes A]	Volume flow (Q) [m^3/s]
Displacement	Displacement (x) [m]	Angle [radian]	Charge (q) [A-s]	Volume [m^3]
Momentum	Momentum [N-s]	Angular Momentum [N-m-s]	Flux linkage [V-s]	Pressure momentum [$N\text{-}s/m^2$]

Bond Graphs and Block Diagrams



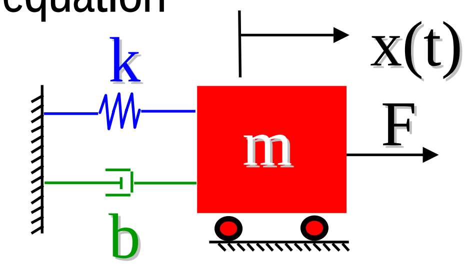




Simple Oscillator Differential Equations
Newton's Law approach

$$m\ddot{x} = F - kx - b\frac{dx}{dt}$$

Second order differential equation

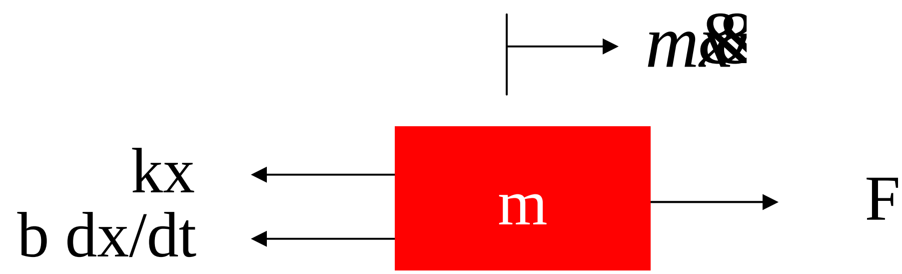


$$m\ddot{x} + b\frac{dx}{dt} + kx = F$$

Let

$$x_1 = x$$

$$x_2 = \frac{dx}{dt}$$



Then

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = \frac{1}{m}(F - kx_1 - bx_2)$$

Second order differential equation
Represented as two first order differential equations
State Space Form

Simple Oscillator State Space Form

Defining u = F and the outputs as states, we get in a state space form

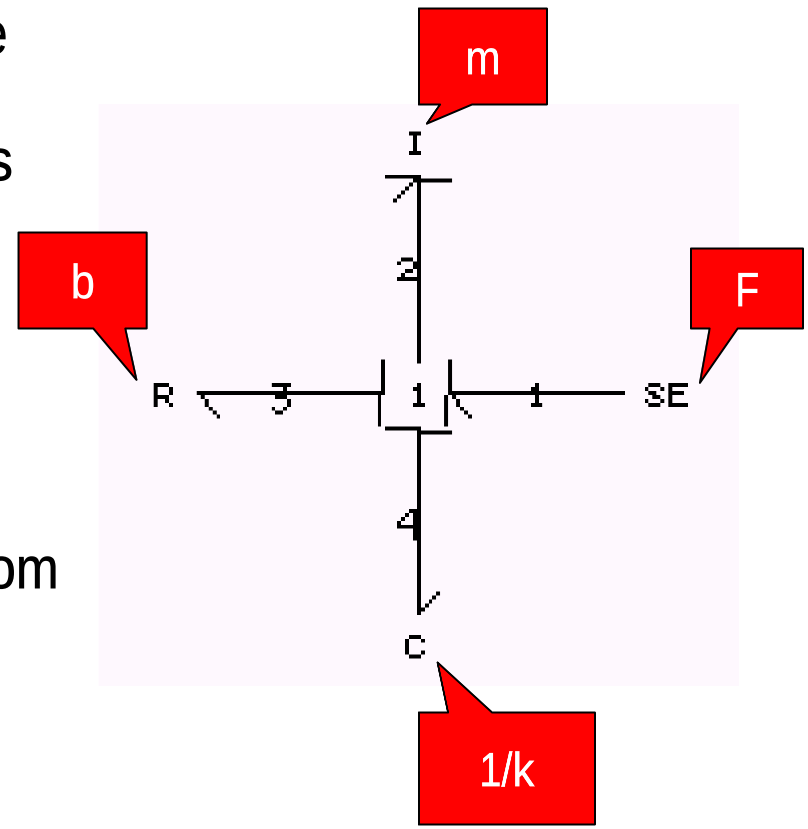
$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

Simple Oscillator Bond Graph Model

Using Inertia, Compliance and resistive elements, the physical system is modeled as shown. Here the I's and C's each produce first order differential equations.

In the bond graph model shown, the index for momentum (p) and displacement (q) variables are taken from the bond number in the figure. For example Inertia [I] connects to the 1 junction through bond 2.



Power Flow in the Simple Oscillator

Simple Oscillator Differential equations derived from the Bond Graph

State Variables are q4 and p2
q4= spring deformation
p2= momentum of mass [I]

$$\frac{dq_4}{dt} = f_4 = f_3 = f_2 = \frac{1}{I} p_2$$

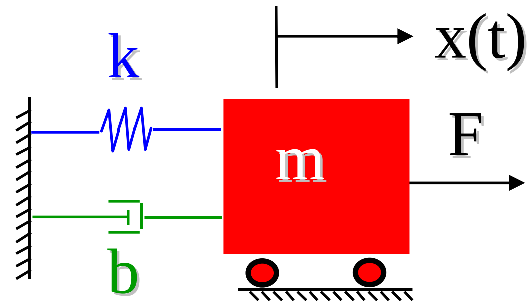
$$\frac{dp_2}{dt} = e_2 = e_1 - e_3 - e_4$$

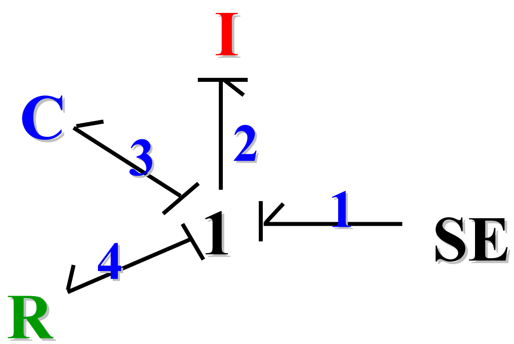
Substituting

$$\frac{dp_2}{dt} = SE - R\frac{dq_4}{dt} - \frac{1}{C} q_4$$

yielding

$$\frac{dp_2}{dt} = SE - \frac{R}{I} p_2 - \frac{1}{C} q_4$$



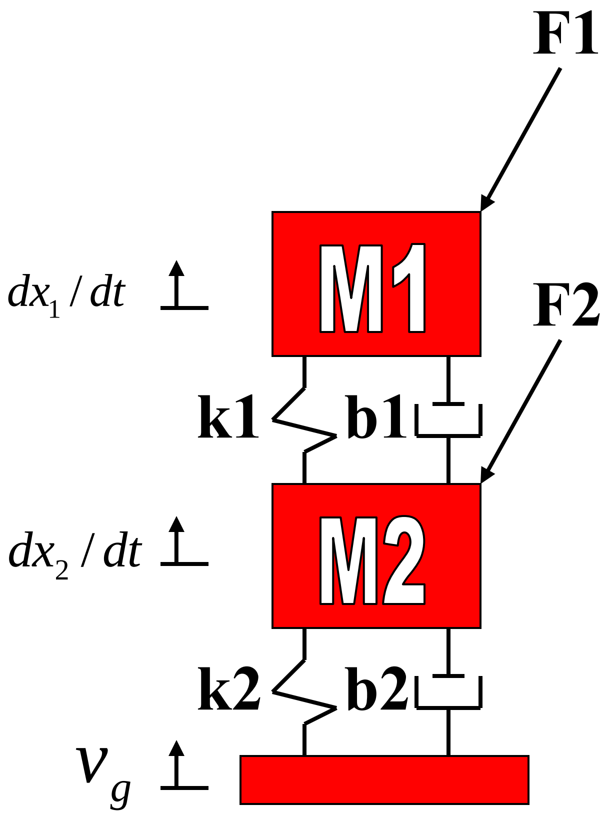


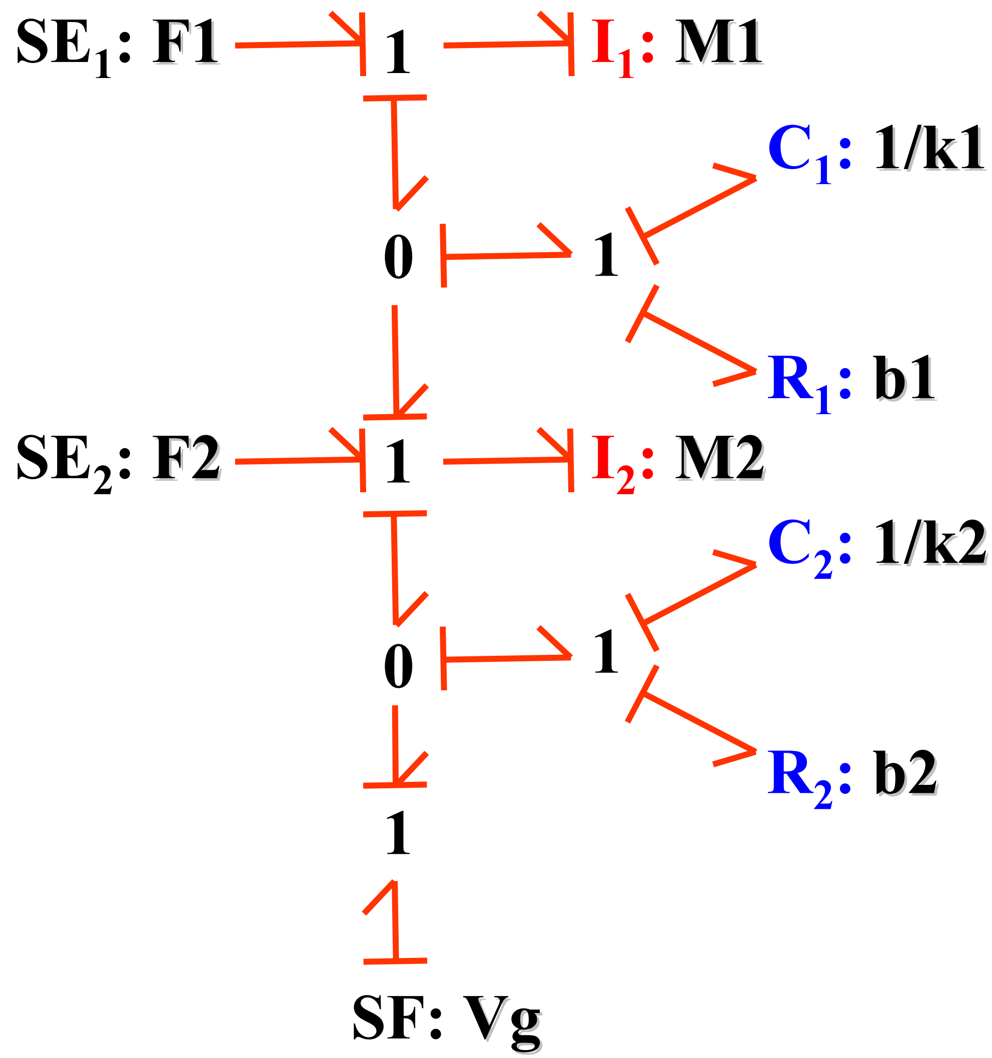
Simple Oscillator State Space equations from the Bond Graph Model

$$\begin{bmatrix} \dfrac{dq_4}{dt} \\ \dfrac{dp_2}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1/I \\ -1/C & -R/I \end{bmatrix} \begin{bmatrix} q_4 \\ p_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} SE$$

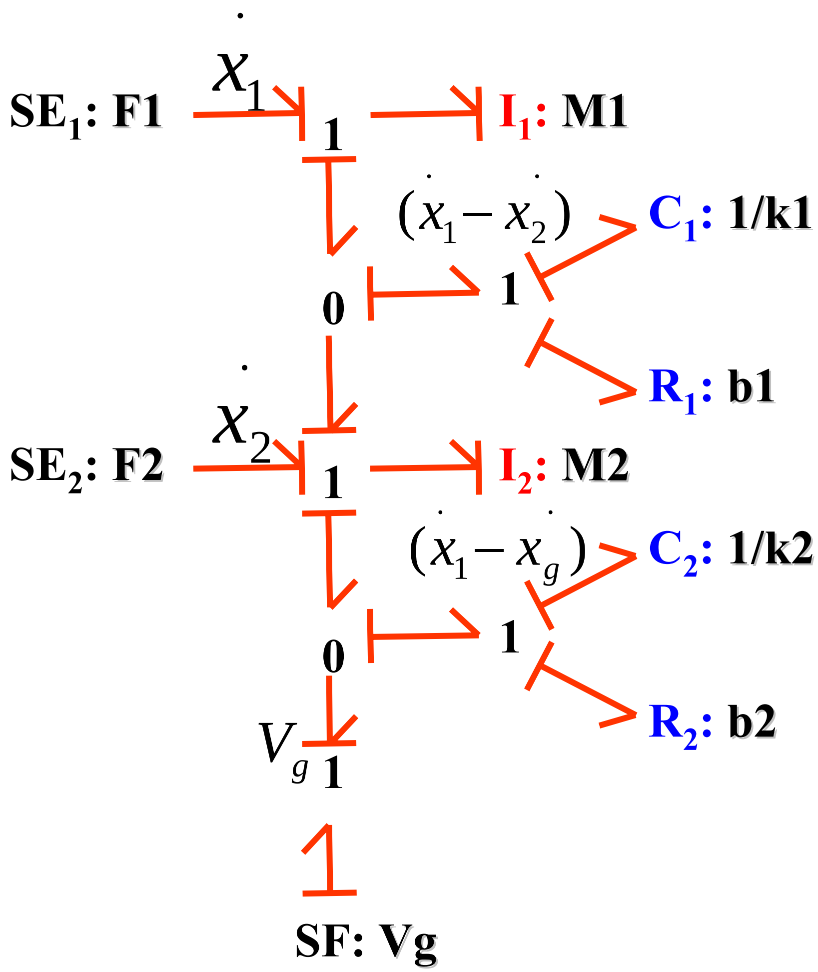
$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} q_4 \\ q_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1/m \end{bmatrix} \begin{bmatrix} q_4 \\ p_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} F$$

Two Mass Model

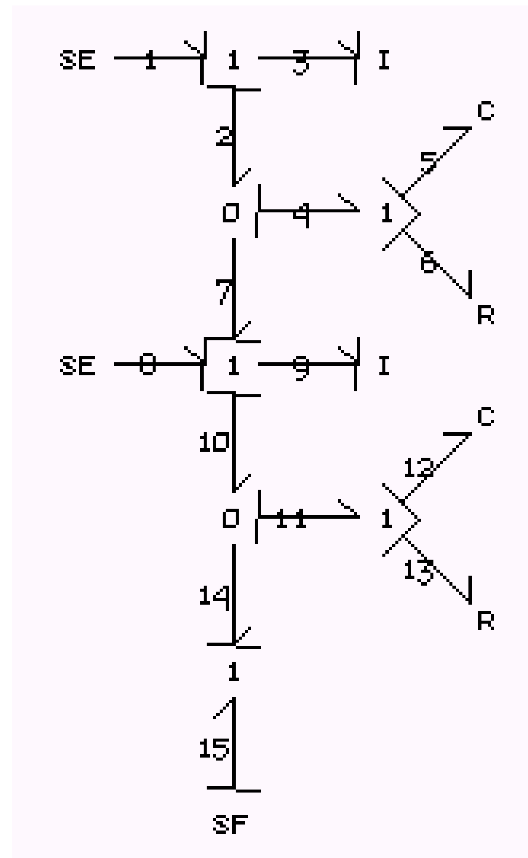




Bond Graph Model



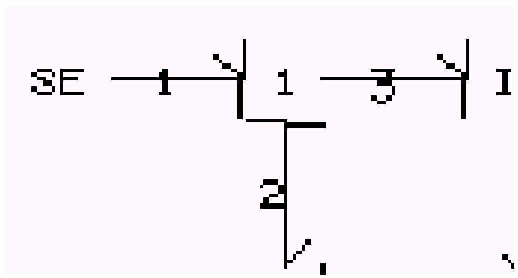
Velocity relations at 1 junctions
Summation of Forces at 0 Junctions



CAMPG BOND GRAPH

Basic Equations, Power Flow and Causality

$$SE_1 = -m_1 g$$
$$I_3 = m_1$$
$$C_5 = 1/k_1$$
$$R_6 = b_1$$
$$I_9 = m_2$$
$$SE_8 = -m_2 g$$
$$C_{12} = 1/k_2$$
$$R_{13} = b_2$$
$$SF_{15} = v_g$$
$$e_1 = e_2 + e_3$$
$$f_1 = f_3 = f_2$$



Example

<u>System Description</u>

Power Flow

Bond	From	To
1	SE_1	1_1_2_3
2	1_1_2_3	0_2_4_7
3	1_1_2_3	I_3

Causality Flow

Bond	From	To
1	SE_1	1_1_2_3
2	0_2_4_7	1_1_2_3
3	1_1_2_3	I_3

Equations Assembly

Equations at the first O junction are

$$e_4 = e_5 + e_6$$

$$f_4 = f_5 = f_6$$

Equations at the second 1 junction are

$$f_2 = f_4 + f_7$$

$$e_2 = e_4 = e_7$$

Continuing to the next junction

$$e_8 + e_7 = e_9 + e_{10}$$

$$f_7 = f_8 = f_9 = f_{10}$$

Differential Equations

Equation (1)

$$dq_5/dt = f_5 = f_4$$
$$dq_5/dt = f_2 - f_7$$
$$dq_5/dt = f_3 - f_9$$
$$dq_5/dt = (1/I_3)p_3 - (1/I_9)p_9$$

Equation (2)

$$dq_{12}/dt = f_{12} = f_{11}$$
$$dq_{12}/dt = f_{10} - f_{14}$$
$$dq_{12}/dt = f_9 - f_{15}$$
$$dq_{12}/dt = (1/I_9)p_9 - SF_{15}$$

Equations (3 and 4)

$$dp_3/dt = SE_1 - (1/C)q_5 - (R_6/I_3)p_3 + (R_6/I_9)p_9$$
$$dp_9/dt = q_5/C + (R_6/I_3)p_3 - (R_6/I_9)p_9 + SE_8 - q_{12}/C_{12} - (p_9/I_9 - SF_{15})R_{13}$$

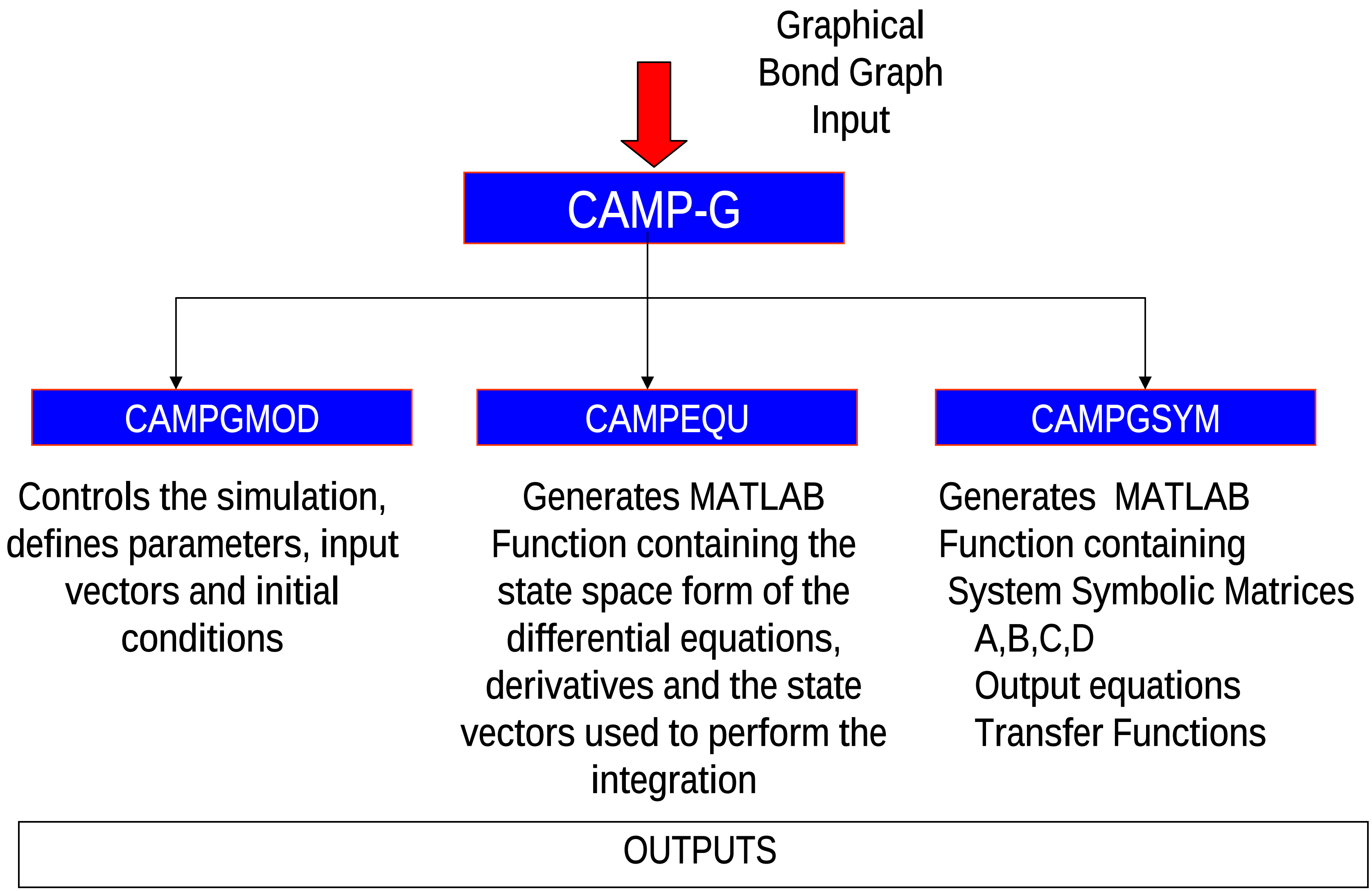
Differential Equations (Classical Approach)

$$m_1 \ddot{x}_1 + b_1 \frac{dx_1}{dt} - b_1 \frac{dx_2}{dt} + k_1 x_1 - k_1 x_2 = -m_1 g$$

$$m_2 \ddot{x}_2 - b_1 \frac{dx_1}{dt} + (b_1 + b_2) \frac{dx_2}{dt} - k_1 x_1 + (k_1 + k_2) x_2 = b_2 v_g - m_2 g$$

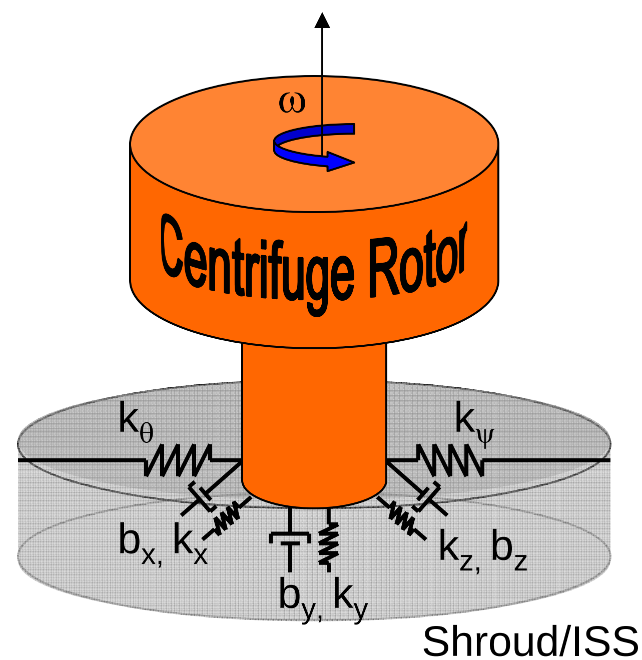
These equations can be converted to the first order state space form using standard approaches.

Bond Graph Modeling – Graphical Tools

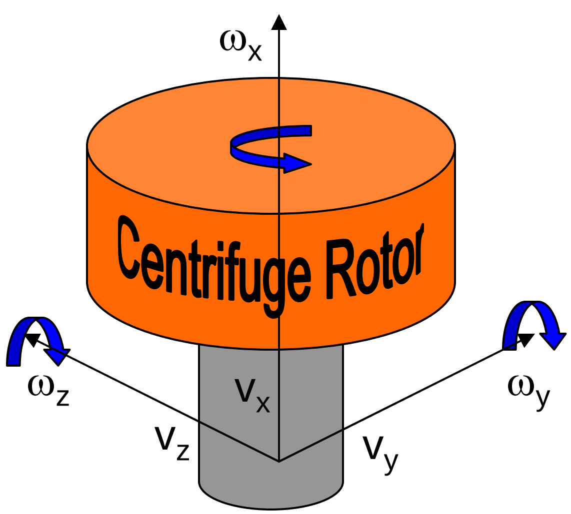


ISS Centrifuge Rotor

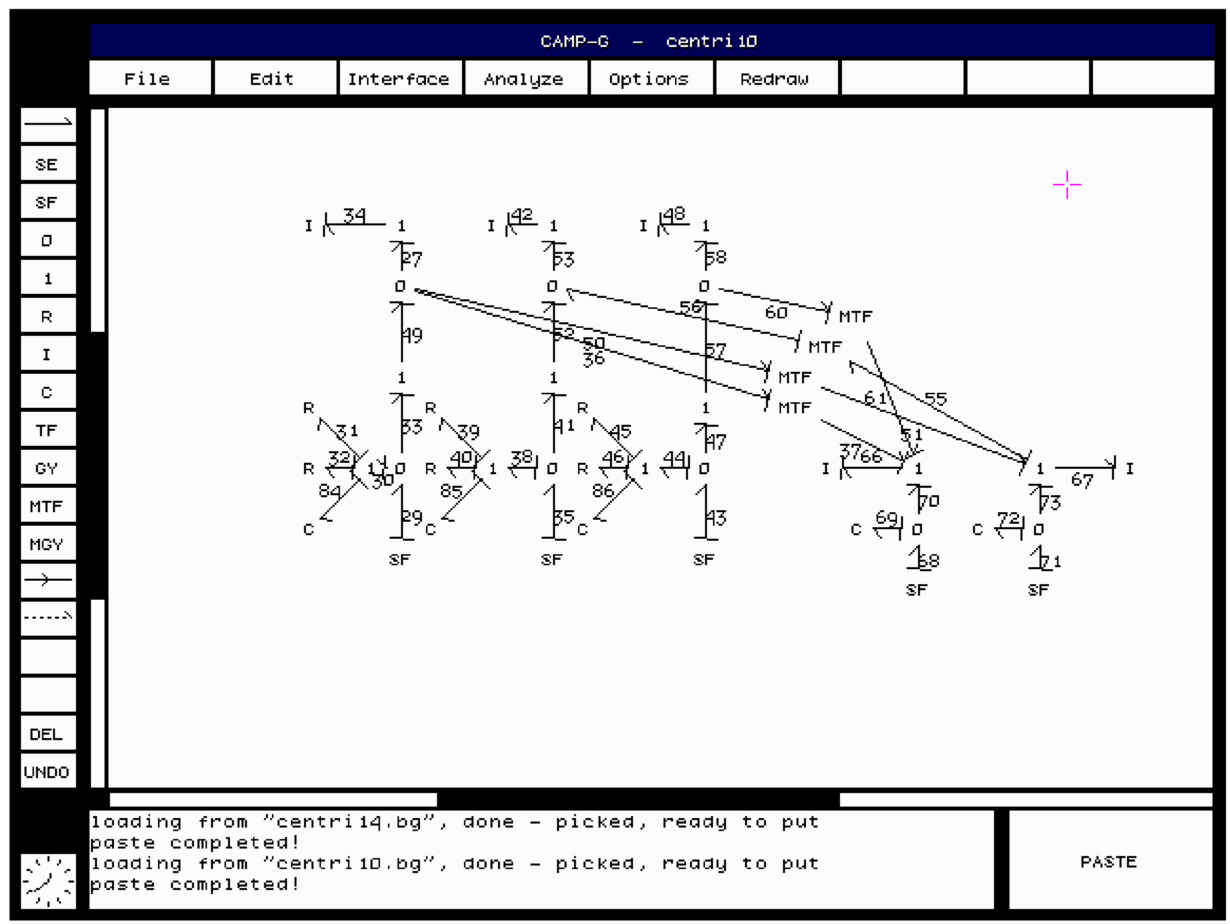
- Simple model depicted as a 5 Degree-of-freedom (10 first order equation) system.
- Vibration isolation system consisting of springs and dampers for the translational motion (3 dof), and rotational springs (2-dof) for the tilting motion of the stator



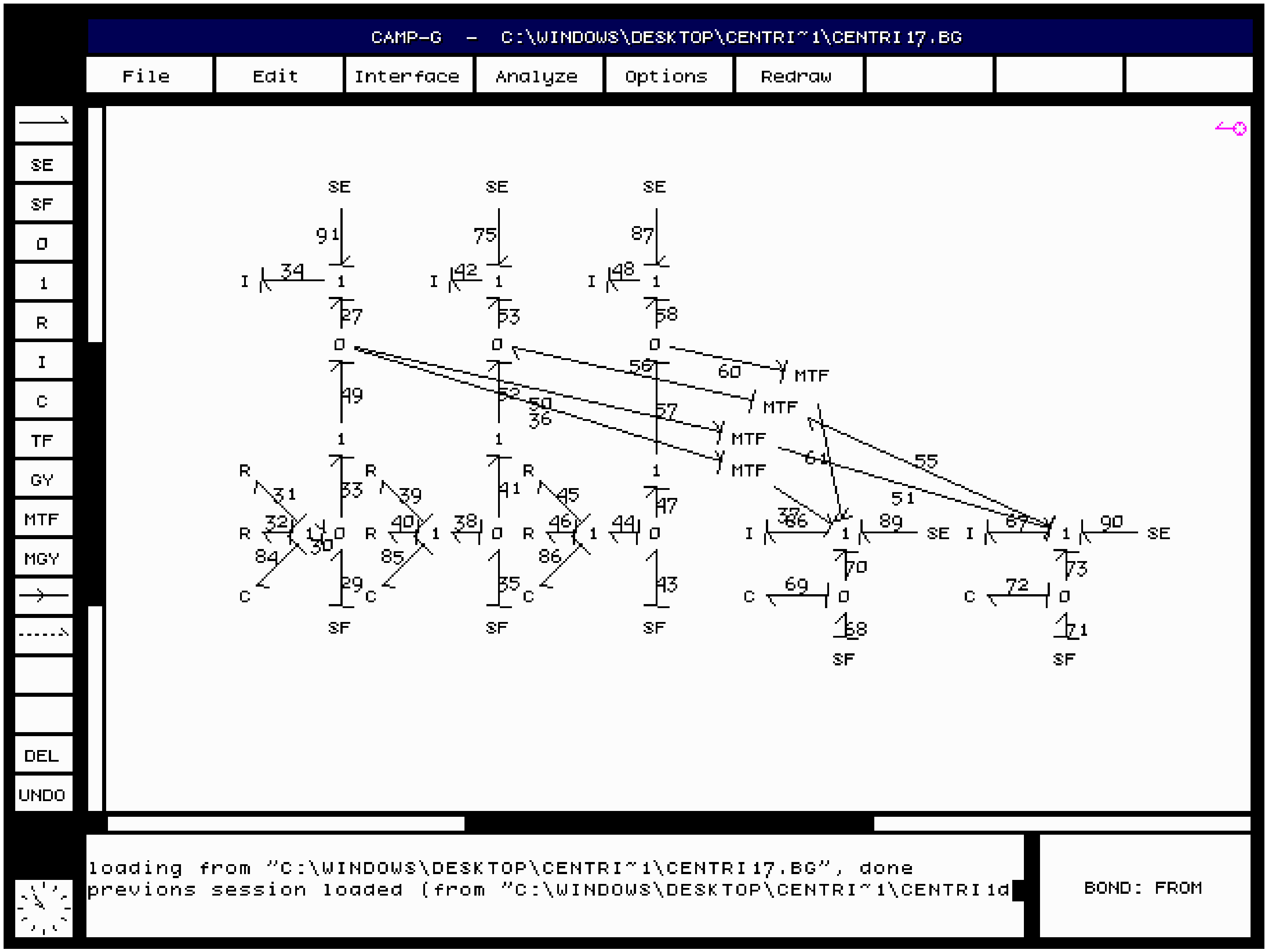
ISS Centrifuge Rotor



Bond Graph computer Model



ISS Centrifuge Rotor External excitation forces and torques



CAMPGMOD (Sample)
Define Physical Parameters

```
%   CAMPG/MATLAB - GENERATED MODEL DESCRIPTION:
%    The following files have been generated
%    campgmod.m  => m file containing model parameters
%                intial conditions, sources and simulation controls
%    campgequ.m  => m function containing the system
%                first order differential equations
%    campgsym.m  => m file containing system matrices in
%                symbolic form
%
%   For simulation and control, edit these files
%    Enter values for physical parameters, initial
%    conditions,inputs and time controls
%    in places where the ?? marks appear
%    Standard generalized variables in Bond Graph notation used.
%
%......CAMPGMOD.M - MATLAB MODEL INPUT FILE  ......
  clear
  more on
% ......System Physical Parameters........
  global R31 R32 I34 T36x37 R39 R40 I42 R45 R46 I48 T50x51  ...
  T55x56 T60x61 I66 I67 C69 C72 C84 C85 C86

% ......  Initial conditions  ........
            Q69IN= 0 ;    % Initial angular displacement about y axis
            Q72IN= 0 ;    % Initial angular displacement about z axis
            Q85IN= 0 ;    % Initial displacement along y axis
            Q86IN= 0 ;    % Initial displacement along z azis
            Q84IN= 0 ;    % Initial displacement along x axis
            P66IN= 0 ;    % Initial angular momentum about the y axis
            P67IN= 0 ;    % Initial angular momentum about the z axis
            P42IN= 0 ;    % Initial linear momentum along the y axis
            P48IN= 0 ;    % Initial linear momentum along the z axis
            P34IN= 0 ;    % Initial linear momentum along the x axis
  initial = [Q69IN; Q72IN; Q85IN; Q86IN; Q84IN; P66IN; P67IN;  ...
  P42IN; P48IN; P34IN] ;
 % ......System Physical Parameters........
  global R31 R32 I34 T36x37 R39 R40 I42 R45 R46 I48 T50x51  ...
  T55x56 T60x61 I66 I67 C69 C72 C84 C85 C86
% Position parameters.  Two angles phiy, phz
```

CAMPGEQU (Example) Solution of linear or nonlinear differential equations

```
% Forcing Function for the 5 DOF Case
% Modeled after CR Simulation Report - Murugan - Equations  _------
%
% Initialization
%
% These are TEST numbers only - For actual model use the w profile from the
% spin up and calculate wdot from the same model; Also update values for
% alpha, gamma and epsilon; Beta is as selected in the report.
%
beta= 45*2*pi/360; %  Radians
%w=0.1 ;  %  rad/s
%wdot= 0.01;  %  rad/(s)(s)
Jxy = 1; % Inertia (TBU)
Jxz = 1; % Inertia (TBU)
Ip = 1; % Inertia (TBU)
It = 0.1; % Inertia (TBU)
gamma = 30*2*pi/360;
alpha = 1;
M = 1404; % Mass Rotor-Stator
e= 0.01; % Epsilon Parameter
%
% Force and Torque Computation
%
% Inside the logical if loop to account for change
%
%   Apply Forces and Torques to the 5 DOF CR Model
%
%Fvector= [Fx Fy Fz Tx Ty Tz]';
% Defining the input of the angular velocity with an acceleration
% during the first 40 seconds.
% md=1074.8 ; % Kg (disturbance mass)
% e= (0.01233/3.28);  % meters  exentricity
```

```
if t <= 0
   w=0;
   SE75= 0 ; % newtons
   SE87= 0 ; %newtons
   SE89= 0 ;
   SE90= 0 ;
   SE91= 0 ;
 elseif t >= 0 & t <= 40
   w=(0.7/40*t)*2*pi; % (radians/sec)
   wdot=.7/40;
   Fx = 0; % Force Units
   Fy = M*e*(w^2*cos(beta+w*t)+wdot*sin(beta+w*t)) ;% Force Units
   Fz = M*e*(w^2*cos(beta+w*t)+wdot*sin(beta+w*t)) ;% Force Units
   Tx = 0; % Torque units
   Ty = -Jxy*wdot+(Ip-It)*alpha*w^2*cos(gamma);
   Tz = -Jxz*wdot+(Ip-It)*alpha*w^2*sin(gamma);
   SE75= Fy ; % newtons
   SE87= Fz ; %newtons
   SE89= Tz ;
   SE90= Ty ;
   SE91= Fx ;
 else
   w=(0.7)*2*pi;  % (radians/sec) constant
   wdot=0 ;
   Fx = 0; % Force Units
   Fy = M*e*(w^2*cos(beta*w*t)+wdot*sin(beta*w*t)) ;% Force Units
   Fz = M*e*(w^2*cos(beta*w*t)+wdot*sin(beta*w*t)) ;% Force Units
   Tx = 0; % Torque units
   Ty = -Jxy*wdot+(Ip-It)*alpha*w^2*cos(gamma);
   Tz = -Jxz*wdot+(Ip-It)*alpha*w^2*sin(gamma);
   SE75= Fy ; % newtons
   SE87= Fz ; %newtons
   SE89= Tz ;
   SE90= Ty ;
   SE91= Fx ;
 end
```

CAMPGSYM (Example) System Matrices

- % System Differential Equations-First Order Form
- % Derivatives vector
- % p_qdot = [dQ69; dQ72; dQ85; dQ86; dQ84; dP66; dP67; dP42; ...
- % dP48; dP34] ;
- % ... System Differential Equations-State Space form A, B, C, D symbolic matrices...
- %...Derivatives (dp,dq) and output variables (e,f)...
- %
- %... Generate A, B matrices corresponding to states p'sand q's ...
- % dQ69=SF68-P66/I66
- %... Number of States 10
- A(1,:) = [sb,sb,sb,sb,sb,sb,sb,sb,sb,sb];
- A(1,:) = [0,0,0,0,0,-1/I66,0,0,0,0];
- B(1,:) = [sb,sb,sb,sb,sb,sb,sb,sb,sb,sb];
- D(1,:) = [sb,sb,sb,sb,sb,sb,sb,sb,sb,sb];
- B(1,:) = [0,0,0,1,0,0,0,0,0,0];
- % dQ72=SF71-P67/I67
- A(2,:) = [0,0,0,0,0,0,-1/I67,0,0,0];
- B(2,:) = [0,0,0,0,1,0,0,0,0,0];
- % dQ85=SF35-P42/I42+P67/I67*T55x56
- A(3,:) = [0,0,0,0,0,0,+1/I67*T55x56,-1/I42,0,0];
- B(3,:) = [0,1,0,0,0,0,0,0,0,0];
- % dQ86=SF43-P48/I48-P66/I66/T60x61
- A(4,:) = [0,0,0,0,0,-1/I66/T60x61,0,0,-1/I48,0];
- B(4,:) = [0,0,1,0,0,0,0,0,0,0];
- % dQ84=SF29-P34/I34-P66/I66/T36x37-P67/I67/T50x51
- A(5,:) = [0,0,0,0,0,-1/I66/T36x37,-1/I67/T50x51,0,0,-1/I34];
- B(5,:) = [1,0,0,0,0,0,0,0,0,0];
-

- % dP66=SF29*R31/T36x37-P34/I34*R31/T36x37- P66/I66/T36x37*R31/T36x37- ...
- % P67/I67/T50x51*R31/T36x37+SF29*R32/T36x37-P34/I34*R32/T36x37- ...
- % P66/I66/T36x37*R32/T36x37-P67/I67/T50x51*R32/T36x37+Q84/C84/T36x37+ ...
- % SF43*R45/T60x61-P48/I48*R45/T60x61-P66/I66/T60x61*R45/T60x61+ ...
- % SF43*R46/T60x61-P48/I48*R46/T60x61-P66/I66/T60x61*R46/T60x61+ ...
- % Q86/C86/T60x61+Q69/C69+SE89
- A(6,:) = [+1/C69,0,0,+1/C86/T60x61,+1/C84/T36x37,-1/I66/T36x37*R31/T36x37- ...
- 1/I66/T36x37*R32/T36x37-1/I66/T60x61*R45/T60x61-1/I66/T60x61*R46/T60x61,- ...
- 1/I67/T50x51*R31/T36x37-1/I67/T50x51*R32/T36x37,0,-1/I48*R45/T60x61- ...
- 1/I48*R46/T60x61,-1/I34*R31/T36x37-1/I34*R32/T36x37];
- B(6,:) = [1*R31/T36x37+1*R32/T36x37,0,+1*R45/T60x61+1*R46/T60x61, ...
- 0,0,0,0,+1,0,0];
- % dP67=SF29*R31/T50x51-P34/I34*R31/T50x51-P66/I66/T36x37*R31/T50x51- ...
- % P67/I67/T50x51*R31/T50x51+SF29*R32/T50x51-P34/I34*R32/T50x51- ...
- % P66/I66/T36x37*R32/T50x51-P67/I67/T50x51*R32/T50x51+Q84/C84/T50x51- ...
- % SF35*R39*T55x56+P42/I42*R39*T55x56-P67/I67*T55x56*R39*T55x56- ...
- % SF35*R40*T55x56+P42/I42*R40*T55x56-P67/I67*T55x56*R40*T55x56- ...
- % Q85/C85*T55x56+Q72/C72+SE90
- A(7,:) = [0,+1/C72,-1/C85*T55x56,0,+1/C84/T50x51,-1/I66/T36x37*R31/T50x51- ...
- 1/I66/T36x37*R32/T50x51,-1/I67/T50x51*R31/T50x51-1/I67/T50x51*R32/T50x51- ...
- 1/I67*T55x56*R39*T55x56-1/I67*T55x56*R40*T55x56,+1/I42*R39*T55x56+ ...
- 1/I42*R40*T55x56,0,-1/I34*R31/T50x51-1/I34*R32/T50x51];
- B(7,:) = [1*R31/T50x51+1*R32/T50x51,-1*R39*T55x56-1*R40*T55x56, ...
- 0,0,0,0,0,0,+1,0];
- % dP42=SF35*R39-P42/I42*R39+P67/I67*T55x56*R39+SF35*R40-P42/I42*R40+ ...
- % P67/I67*T55x56*R40+Q85/C85+SE75
- A(8,:) = [0,0,+1/C85,0,0,0,+1/I67*T55x56*R39+1/I67*T55x56*R40,- ...
- 1/I42*R39-1/I42*R40,0,0];

System Matrices from Bond Graph A(:,1:7)

A(:,1:5) (All Rows, First 5 Columns)

- [0, 0, 0, 0, 0]
- [0, 0, 0, 0, 0]
- [0, 0, 0, 0, 0]
- [0, 0, 0, 0, 0]
- [0, 0, 0, 0, 0]
- [1/C69, 0, 0, 1/C86/T60x61, 1/C84/T36x37]
- [0, 1/C72, -1/C85*T55x56, 0, 1/C84/T50x51]
- [0, 0, 1/C85, 0, 0]
- [0, 0, 0, 1/C86, 0]
- [0, 0, 0, 0, 1/C84]

A(:,6:7) All rows, Columns 6 & 7

- [-1/I66, 0]
- [0, -1/I67]
- [0, 1/I67*T55x56]
- [-1/I66/T60x61, 0]
- [-1/I66/T36x37, -1/I67/T50x51]
- [-1/I66/T36x37^2*R31-1/I66/T36x37^2*R32-1/I66/T60x61^2*R45-1/I66/T60x61^2*R46, -1/I67/T50x51*R31/T36x37-1/I67/T50x51*R32/T36x37]
- [-1/I66/T36x37*R31/T50x51-1/I66/T36x37*R32/T50x51, -1/I67/T50x51^2*R31-1/I67/T50x51^2*R32-1/I67*T55x56^2*R39-1/I67*T55x56^2*R40]
- [0, 1/I67*T55x56*R39+1/I67*T55x56*R40]
- [-1/I66/T60x61*R45-1/I66/T60x61*R46, 0]
- [-1/I66/T36x37*R31-1/I66/T36x37*R32, -1/I67/T50x51*R31-1/I67/T50x51*R32]

System Matrices from Bond Graph A(:,8:10)

A(:,8:10) All rows, Columns 8-10

- [0, 0, 0]
- [0, 0, 0]
- [-1/I42, 0, 0]
- [0, -1/I48, 0]
- [0, 0, -1/I34]
- [0, -1/I48*R45/T60x61-1/I48*R46/T60x61, -1/I34*R31/T36x37-1/I34*R32/T36x37]
- [1/I42*R39*T55x56+1/I42*R40*T55x56, 0, -1/I34*R31/T50x51-1/I34*R32/T50x51]
- [-1/I42*R39-1/I42*R40, 0, 0]
- [0, -1/I48*R45-1/I48*R46, 0]
- [0, 0, -1/I34*R31-1/I34*R32]

System Matrices from Bond Graph B(:,1:10)

B(:,1:5) All rows, Columns 1-5

- [0, 0, 0, 1, 0]
- [0, 0, 0, 0, 1]
- [0, 1, 0, 0, 0]
- [0, 0, 1, 0, 0]
- [1, 0, 0, 0, 0]
- [R31/T36x37+R32/T36x37, 0, R45/T60x61+R46/T60x61, 0, 0]
- [R31/T50x51+R32/T50x51, -R39*T55x56-R40*T55x56, 0, 0, 0]
- [0, R39+R40, 0, 0, 0]
- [0, 0, R45+R46, 0, 0]
- [R31+R32, 0, 0, 0, 0]

Eigenvalues Comparison (6th Order State Space)

	Euler/Newton Approach			Bond Graph Approach		
#	**Eigenvalue**	**ω rad/s**	**ζ Damping Factor**	**Eigenvalue**	**ω rad/s**	**ζ Damping Factor**
1	-0.3960 ± 0.7391i	0.8385	0.4722	-0.3960 ± 0.7391i	0.8385	0.4723
2	-0.2046 ± 0.8273i	0.8522	0.2401	-0.2047 ± 0.8273i	0.8522	0.2401
3	-0.2089 ± 0.8353i	0.8611	0.2426	-0.2089 ± 0.8354i	0.8611	0.2426

Locked rotational degrees of freedom

Eigenvalues Comparison (10th Order State Space)

	Euler-Lagrange Approach			Bond Graph Approach		
#	**Eigenvalue**	**ω rad/s**	**ζ Damping Factor**	**Eigenvalue**	**ω rad/s**	**ζ Damping Factor**
1	-0.3960 ± 0.7391i	0.8385	0.4722	-0.3960 ± 0.7391i	0.8385	0.4723
2	-0.1857 ± 0.8142i	0.8351	0.2223	-0.1838 ± 0.8127i	0.8332	0.2205
3	-0.1888 ± 0.8217i	0.8431	0.2239	-0.1877 ± 0.8208i	0.8419	0.2229
4	-0.1642 ± 3.2507i	3.2548	0.0504	-0.1880 ± 3.3196i	3.3249	0.0565
5	-0.1737 ± 3.2771i	3.2816	0.0529	-0.1885 ± 3.3192i	3.3245	0.0566

Summary

- Bond Graph Method provides an elegant alternative graphical method for modeling dynamical systems.
- Results for complex systems like the Centrifuge Rotor correlate well with other approaches.
- CAMP-G generates MATLAB M-file automatically
  - System Equations
  - System Equations in a symbolic form
  - Transfer Functions
- Easily combine bond graph models with mathematical and analysis capabilities of MATLAB

References

- *Ooi*, 2004, Y. Ooi, “ Centrifuge Rotor (CR) Rotor Control Analysis Report,” CR15-Y02015, Date: 05/10/2004, Issue: A.
- *Murugan, 2005*, Murugan Subramaniam, “Assessment of Joint Springs of Centrifuge Rotor Model,” Memo ATL-05-HM04, August 2005.
- *Granda, 2005,* Jose J. Granda, “Methods for Developing Computer Models of the Space Station Centrifuge Rotor and the Orbiter Repair Maneuvers”, Final Report, NASA Summer Faculty Research Opportunities Program 2005, NASA JSC, August 2005
- *HighTech Consultants*, 2001 Website, “An Introduction to Bond Graphs”, www.bondgraphs.com
- Karnopp, Margolis, Rosenberg. System Dynamics   Modeling and Simulation of Mechatronics Systems. Wiley, 2005